7Р

УДК 004.89

А.И. Разумовский
Институт проблем управления им. В.А. Трапезникова РАН, Россия

Реализация контекста проектирования как обеспечение синтеза структуры алгоритма

В статье рассматриваются ключевые проблемы, возникающие при проектировании сложных программных систем и алгоритмов. Раскрыта фактология стадий проектирования, методологический инструментарий моделирования. Даны практические рекомендации для построения локального контекста проектирования и создания целостной среды моделирования и синтеза структуры алгоритма.

Введение

Проектирование сложных специализированных программных систем является дорогостоящим, трудоемким процессом, не всегда приводящим к положительным результатам. Причиной тому на протяжении всего процесса проектирования программной системы (ПС) является все более возрастающая сложность учета многочисленных дополнений и уточнений отдельных ее элементов и характеристик. На практике для контроля над сложностью элементов системы используются различные формальные и технические методы и инструменты, которые позволяют проводить анализ потенциальных возможностей системы и формировать синтезированные решения, увязывая их с атрибутами целевых требований на разных уровнях описаний: от языковых до графических методологических, моделирующих пространство требований.

Чтобы понять ключевые проблемы, наличествующие при проектировании сложных программных систем и алгоритмов посредством имеющихся средств и инструментов анализа и синтеза программных систем, следует обратить внимание на тот факт, что главным действующим персонажем процесса проектирования остается человек. Иначе говоря, при проектировании необходимо использовать такие методы и инструменты, которые бы наиболее соответствовали корневым особенностям человеческого организма, включая в первую очередь психические свойства, такие, как память и зрение.

Еще на рубеже XIX – XX веков Дж.М. Кэттелл экспериментально установил, что внимание человека может быть одновременно сосредоточено на пяти – максимум семи элементах. В 1956 году Джоном Миллером была опубликована статья «Магическое число семь» [1], которая была посвящена проблеме памяти и написана на основе развиваемой автором информационной теории. Им было показано, что объем кратковременной памяти человека, которая непосредственно используется при поиске, анализе, сравнении и встраивании элементов в модель системы, не превышает числа «семь плюс / минус два» элемента. Также Миллер сумел показать, что люди способны расширить ограниченные возможности кратковременной памяти, группируя отдельные единицы информации и используя символы для обозначения каждой из групп.

Проектирование сложной программной системы, в особенности на ранней его стадии, заключается в сборе необходимой информации о требованиях функционирования алгоритма системы и выборе стратегических решений по реализации этих требований в модели. Среди языковых средств описания требований и целей ПС наибо-

лее популярным является язык UML и UML-2 [2], [3]. Как указывается в [3]: «UML представляет собой язык моделирования, который не только весьма выразителен визуально – графически, но и способен точно и недвусмысленно формально описывать состояния программной системы». Однако, наряду с выразительностью и точностью представления модели системы, для полноты восприятия она должна вмещать в себя качественные и количественные выражения всякого рода взаимоотношений между элементами системы, а также максимально аккумулировать не только ассоциативные данные внутрисистемных связей, но и предоставлять реальные решения для достижения производных целей. Тогда подобная модель будет служить гибким проводником между входным блоком качественных целей системы и формальным их выражением в рамках и при контроле человеческого фактора. Важность ориентации на человеческий фактор диктуется также стремлением отыскивать смысловой базис при абстрагировании от реальной задачи, в стремлении упростить и формализовать трудноописываемые факторы (например, когнитивную информацию). Как указывают в своей работе Т. Пратт и М. Зелковиц: «Выбрав неверную теоретическую модель, можно ее исследовать и получить какие-то результаты, которые, однако, невозможно будет преобразовать в решение исходной практической задачи» [4].

Образование новых знаний и модели памяти

Память человека является корневым психическим образованием, благодаря которому становится возможным накопление сознательного опыта. Главенствующую роль памяти в накоплении знаний отводили еще античные мыслители – Платон, Аристотель и Плотин. Так Аристотель связывает память со временем: «Память же есть память о прошлом, и помнят только те, у которых есть ощущение времени» [5]. Он также различает память и припоминание: «Взятое само по себе, оно есть предмет созерцания и представление, а как относящееся к другому – есть как бы образ и воспоминание» [5]. Связав припоминание с временным фактором, Аристотель указывает на ассоциативный характер процесса познания. Один из последователей Аристотеля – Т. Гоббс в Средние века расширил сущность ассоциативности познания, выразив его в формах «движения образов», из-за которого у человека остается след вещи даже в момент ее исчезновения из поля видимости. Наиболее исторически значимой точкой, оказавшейся в дальнейшем своеобразным мерилом отсчета, стали три ассоциативных принципа Д. Юма, однозначно утверждая память как фактор процесса познания:

– «сходство» – воспринимая один объект, мы мысленно переносимся к другому, похожему;
– «смежность» – упоминание об одном объекте приводит к воспоминанию о других объектах этого же класса;
– «причина и действие» – думая о причине, мы думаем и о следующем за ней действии [6].

В истории изучения памяти как психического явления следует отметить эволюцию нескольких направлений: от работ Спирмена [7], который показал, что в основе памяти лежит общая способность к запоминанию, а сама память характеризуется многофакторной структурой, до исследований А.А. Смирнова [8] и П.И. Зинченко [9], в которых показано огромное влияние способов мнемической обработки на результаты запоминания. Эти материалы свидетельствуют о том, что память представляет собой активный процесс, функционирующий во взаимосвязи с другими процессами, в первую очередь с мышлением.

В западной науке исследования памяти проводились в основном в рамках когнитивной психологии. Возникло множество концепций и моделей памяти, большинство из которых опирается на двух-трехкомпонентную ее основу. Одной из наиболее известных трехкомпонентных моделей памяти, принятой в качестве базовой большинством исследователей, является модель Р. Аткинсона [10]. Согласно его подходу информация из внешней среды попадает первоначально в соответствующее сенсорное хранилище, названное «сенсорный регистр», затем либо стирается, либо переводится в кратковременное хранилище, где сохраняется на десятки секунд в форме вербально-акустического кода. Кратковременная память, по мнению Аткинсона, – это активно функционирующая система. Управляя процессом трансляции информации между блоками, можно удлинить срок хранения информации. Ключевыми для кратковременной памяти являются активные процессы: вербализация материала, перекодирование, принятие решения, выбор способа запоминания, поиск в памяти, повторение и другие (рис. 1).

Признание модели Р. Аткинсона объясняется тем, что с ее помощью удалось теоретически обобщить «множество феноменов памяти, внимания и восприятия, причем сама она прямо воспроизводила архитектуру компьютера: три вида памяти соответствуют интерфейсу, активному процессору и пассивной внешней памяти, а процессы управления – программным алгоритмам, определяющим движение и характер преобразований информации от поступления на вход системы до выдачи ответа» [11].

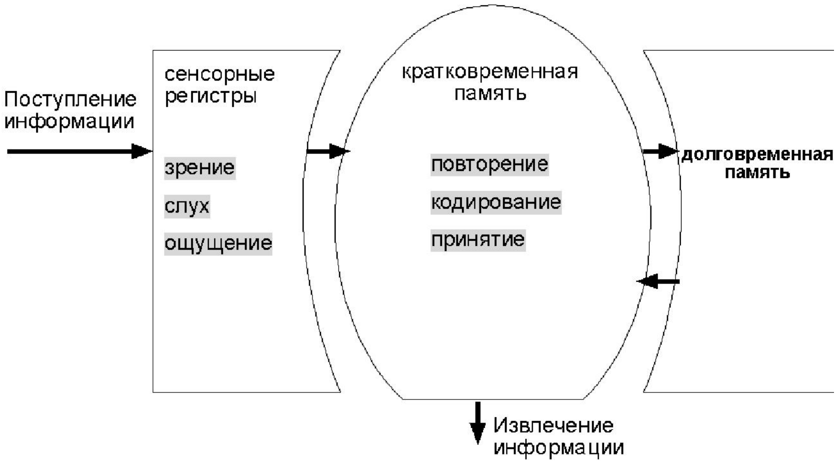


Рисунок 1 – Кратковременная память – узкое место восприятия

Согласно модели Р. Аткинсона, готовность памяти, т.е. способность своевременно актуализировать так называемые «мнемические следы» и воспроизводить необходимую информацию, обеспечивается тем, что одно и то же событие или знание может быть представлено более чем в одной структуре памяти и посредством более чем одного кода (перцептивного, семантического и др.), что повышает вероятность извлечения знания [11]. Для поиска и реализации оптимального пространства проектирования ПС важно отметить следующее: информация, содержащаяся в кратковременной памяти, непосредственно доступна для воспроизведения.

Фактология начальной стадии проектирования алгоритма

Для начальной стадии проектирования алгоритма характерным является когнитивно-качественное описание системы, которое состоит преимущественно из требований функциональности и декларации границ деятельности системы. Важно отметить, что для начального этапа следует в отличие от методологии UML не создавать катего-

риальных определений и выводов для группирования элементов задачи в предопределенные наборы, а наоборот, стремится сохранить максимальную полноту качественного описания элементов алгоритма. Стратегически необходимость сохранить максимальную полноту качественного описания обосновывается желанием контролировать взаимоотношения элементов структуры алгоритма средствами кратковременной памяти человека, которая более восприимчива к конкретным узнаваемым деталям, нежели к абстрактным образам. Между тем, конечным результатом начальной стадии проектирования должен стать каркас системы в ее абстрактном выражении. Следовательно, единственным способом соединить два вышеотмеченных свойства системы является создание комплексных составных элементов, каждый из которых имеет, как минимум, двойное выражение: качественное, в виде сохраненного описания требований и ограничений, и абстрактное, в виде декларативного вида структур данных и функциональных решений.

Обобщая, таким образом, требования к содержанию процесса проектирования, его цели можно свести к следующим позициям:
– необходимость сохранения формулировок когнитивного, эмпирического и качественно-описательного характера; выделение категорий при этом производить специализированно «под задачу» и оформлять в виде условных предложений;
– возможность описания частичной процедурной последовательности алгоритма без обязательного формирования всей функциональной его схемы либо хаотического множества функциональных элементов задачи, когда их древовидная, циклическая, параллельно-независимая связность не выражена явно и непосредственно;
– стремление к сокращению избыточности элементов системы, исключение дублирования функциональных и информационных свойств.

Значительно лучшую управляемость процессом проектирования удастся достичь при сочетании указанных выше требований плюс технический инструментарий для осуществления контроля над их исполнением.

Способ объединения качественной, технической и структурной части знания определяет контекст проектирования (КП). Кратко говоря, КП служит для контроля над качественно-количественными переходами описания деталей алгоритма, а также для сокращения пути доступа между элементами алгоритма.

Методологический инструментарий моделирования

Проектирование сложной программной системы – это постепенный и итеративный процесс, поэтому выработка и использование правильной методологии проектирования вносит в процесс разработки соответствующую организованность.

Упорядоченность разработки в ясно осознаваемой перспективе объединения качественной, технической и связующей частей должна базироваться на свойствах используемой в проекте информации. Следует выделить несколько основных характеристик информации для проектирования при помощи КП: а) технические требования, б) предположения о связях между выявленными элементами, в) предположения о существовании невыявленных элементов, г) вторичные элементы – результаты и следствия.

При рассмотрении способа поступления новых данных в систему важно разграничить, что имеется уже существующая информация и информация вносимая. По отношению к имеющимся данным новые данные неструктурированы, так как неявны, разрознены или не соответствуют имеющимся структурам данных, характеризуемых множеством способов кодирования.

Следовательно, для организации встраивания нового элемента данных в структуру алгоритма, что характеризуется поиском и реализацией необходимых связей между имеющимися и новым элементом, следует качественно и количественно оценить вносимую им информацию. Такую оценку можно провести лишь при наличии соответствующих инструментов, которые позволили бы те или иные элементы информационного потока квалифицировать, придать им определенную значимость – а это и будет свидетельствовать о том, что между элементами информационного пространства, имеющимися и вновь поступающими, способна установиться градация – данные будут выстроены в определенной последовательности, по соответствующему рангу. И такие данные тем самым будут сбалансированы и упорядочены в соответствии с теми признаками, которые были выбраны в начале процесса встраивания нового элемента. Например, по СК. А затем при сравнении друг с другом уже ранжированные элементы могут быть встроены непосредственно в канву имеющихся данных.

Также при встраивании нового элемента стоит вопрос о квантификации или «измерении» вносимой им информации. Тогда операция сравнения приобретет расчетный смысл и может быть осуществлена автоматически.

Следует также отметить, что квалификация данных может быть проведена с помощью инструментов, которые позволяют провести оценку и квалификацию новой информации. Эти инструменты служат как для декомпозиции, в частном случае в виде парсеров, так и для расчета критериальных весов, которые могут использоваться при построении иерархии зависимостей и приоритетных элементов данных.

Как выбираются критерии оценки неструктурированной информации. Очевидно, что при разного рода наборе критериев результирующая иерархия информационных элементов может быть различна. Но когда говорится о встраивании информационного элемента в имеющийся информационный банк, мы обязаны рассчитывать на совпадение критериальной основы, то есть о приведении в соответствие прежде существовавших связей в структуре алгоритма со связями, образующимися при встраивании нового элемента.

В этом смысле удобно соотнестись с таким понятием, как отображение, когда определенному значению ключевого слова или фразы поставлен в соответствие некий образ или обозначение. Этот пример хорошо иллюстрируется при проектировании алгоритмов с использованием ограничивающих, как в языке HTML, или категоризирующих тэгов.

$$\Gamma = (\overline{T}, \overline{V}) \text{ или } \Gamma = (\overline{T}, \overline{V}, \overline{T'}).$$

Каждое локальное отображение будет характеризовать связь между наиболее близкими – качественно, терминологически, семантически – элементами проекта алгоритма. Следовательно, главным стержнем контекста проектирования оказывается множество локальных связей. Тогда элементом КП становится пара (t_i, v_i). Связанные в граф подобные пары необходимо поставить в отношения минимального пути доступа между элементами.

Достижимость между элементами псевдоорграфа должна быть взаимной, а путь между элементами графа КП – минимальным. То есть псевдоорграф КП – есть сильный граф.

Таким образом, обоснование методологии применения КП как среды расширенного восприятия пространства проектирования алгоритма заключается в необходимости сохранять качественный и количественный аспект связности отдельных элементов алгоритма, что невозможно, с одной стороны, без перечисления всех связей как группы элементов с ограниченными критериями связности, с другой – без контроля и согласования их по основному выбранному критерию, например, по способу кодирования или форме представления, с третьей стороны – без обеспечения встраивания в КП результатов операций.

Практика построения контекста проектирования

Контекст проектирования представляет собой модель восприятия человека. Восприятие может состоять из центра восприимчивости – центрального элемента, на который обращено непосредственное внимание проектировщика, угловой (или боковой) информации, число элементов которой варьируется в соответствии с так называемым «магическим числом» Миллера (7 +/– 2), а также невидимых элементов, непосредственно или косвенно связанных с центральным и возникающих в поле зрения при необходимости.

В рамках модели важно проследить переходы между центральным элементом КП и зависимыми членами, изучить количество угловых членов локального контекста с точки зрения оптимального восприятия человеком, а также определить формы представления элементов локального контекста.

Известно, что фундаментальное правило технологии COM выглядит так: «Если Вы смогли попасть куда-то хоть откуда-нибудь, Вы можете попасть туда откуда угодно» [12].

Аналогичным образом можно представить базис КП – каждый элемент алгоритма имеет полное знание о любом смежном элементе.

Совокупность идей и отношений, способствующих реализации КП, может быть сведена к следующим принципиальным позициям:

1) декларация возможности контроля элементов и связей между ними;

2) пространство элементов, множество которых связано между собой отношением эквивалентности, то есть элементы однородны. Одновременно среди равных элементов выделяется один центральный. В разный момент времени в одном множестве центральным может быть выбран произвольный элемент;

3) КП состоит из локальных контекстов проектирования (ЛКП), каждый из которых представляет собой связку из не менее двух элементов – центрального и зависимого, по примеру ассоциации. Таким образом, зависимый будет находиться в контексте ключевого элемента и, следовательно, по одному мы можем знать о другом, или, иначе говоря, произвести вывод, объединяющий элементы в локальный контекст проектирования (ЛКП).

Объединяющий вывод может быть выполнен либо в виде качественно-количественного отношения, либо в виде описаний требований к подзадаче, либо абстрактно: в виде формульного выражения Y = F(X). Итак, происходит взаимосвязь между количественно-абстрактным выражением и качественно-описательным в виде технических требований, пожеланий, примечаний, комментариев. Такая связь выстраивается, очевидно, сквозь четко зафиксированные ЛКП – в частном случае двойки элементов, – содержащие явные переходы – преобразования между двумя описаниями в форме фраз или слов. Примером такого ЛКП может служить ассоциативный массив, в котором имеется безусловный переход от описателя к определителю.

Последовательность составления КП состоит из следующих этапов.

1. Группирование элементов, включающее их анализ, с созданием хранилища.

2. Выделение базовых элементов проектирования на основе сортировки имеющихся в хранилище элементов. Некоторых элементов, таких, как имена функций или квалификаторов, может быть достаточно много, поэтому уместно выбирать в качестве базовых элементов те, которых в проекте наибольшее число.

3. Построение сочетаний из сортированных элементов проектирования, помещение их в главное хранилище, устроенное по ассоциативному принципу, хотя это приведет к нарушению взаимодостижимости вершин на графах ЛКП. Результатом этого этапа будут созданные ассоциативные пары и «звезды» (рис. 2).

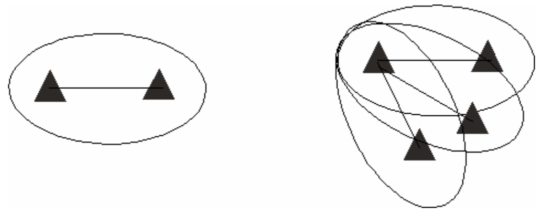


Рисунок 2 – Организация локальных контекстов

Важно, что каждый из элементов проектируемого алгоритма должен включать также еще, как минимум, один, с ним связанный. Следовательно, при соединении в алгоритме или в объекте каждый такой элемент будет содержать от 1 до N квази-смежных (то есть опосредованно- или прямосочлененных) элементов ЛКП.

Минимальным ЛКП может быть «двойка» или пара: $A \leftrightarrow B$.

Примерные качественные значения «двойки» или пары ЛКП таковы:

– А определяется В,
– А исходит из В,
– А ассоциируется с В,
– А следует из В,
– А комментарий к В,
– А параметр в В.

Следует добавить, что, например, в объектном представлении ЛКП могут быть реализованы в виде объектов, содержащих два и более информационных поля. Однако при их использовании выясняется, что восприятие при проектировании алгоритма фокусируется лишь на содержании отдельных его частей, объектов. При этом концентрации внимания на соединения и взаимозависимости между элементами не происходит, иначе говоря, целостных данных о структуре алгоритма достичь не удастся.

4. Расширение КП происходит за счет помещения в 1-ое хранилище нового элемента (СК хранилища – строка, в которой записывается в базу как идентификатор и квалификатор, так и произвольное наименование в естественноязыковой форме). Затем новый элемент ставится в зависимость от уже имеющегося в базе элемента с образованием пары или, если зависимость более сложная, «звезды» и далее помещается в главное хранилище КП.

5. Присоединение результатов к КП диктуется необходимостью расширения алгоритма или агрегации его в иную алгоритмическую структуру. В самом деле, технически базируя присоединение результата к КП на пункте 4, следует учесть исходную фиктивность данного элемента, то, что его не существует изначально, он лишь появится позднее по мере совершения операции. Следовательно, планируя использование результата в КП, формируется отдельная группа результатов под именами «Результат1», «Результат2»,.. «РезультатN», которая встраивается в КП в соответствии с пунктом 4. В дальнейшем, при получении определенных результатов, каждый из них занимает поле вместо (или вместе, в паре) определенного «Результата».

В общем случае: $F(x) = R$ граф ЛК выглядит, как показано на рис. 3. При удалении R, граф стягивается к исходной паре: $F \leftrightarrow x$.

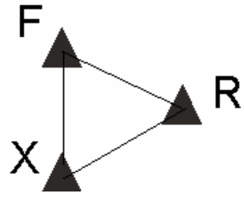


Рисунок 3 – Качественное равенство исходных данных и результатов

Следствия использования КП.

1. Единая база ассоциативных элементов алгоритма.

2. Легкоосуществляемый доступ произвольного элемента в КП или в ЛКП. Таким образом, каждый элемент КП может получить не только комментарий, что используется повсеместно, но и любой иной элемент, находящийся с данным в одном контексте проектирования.

3. Легкое изменение главного элемента в КП позволяет осуществлять навигацию по дереву КП. Главный элемент в этом аспекте применения является переменным ЛК, на который осуществляется реальная (мгновенная) фокусировка внимания.

4. Возможность создания несбалансированных групп на основе введения и контроля весов приоритетов отдельных элементов КП. Приоритеты могут обозначать, например, частоту востребованности (или наоборот, периферийности) элемента или группы.

Накладывая структуру контекста на структуру алгоритма, можно увидеть широкие возможности реализации алгоритма через КП (рис. 4).

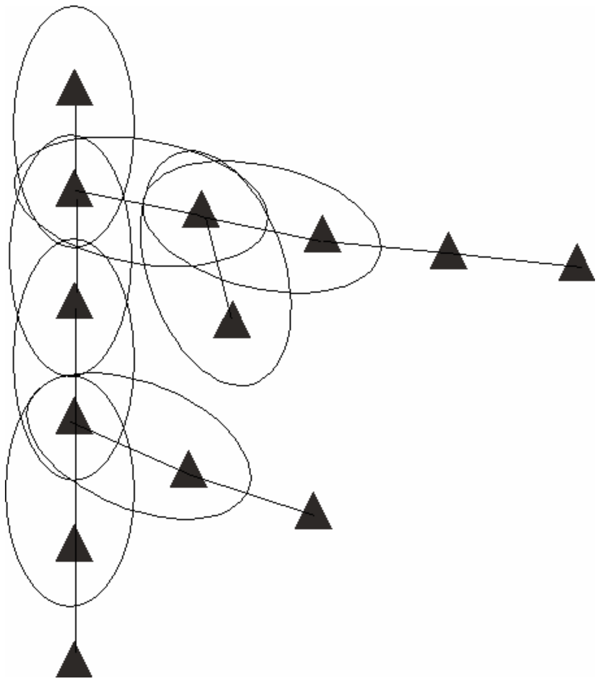


Рисунок 4 – Рассмотрение структуры алгоритма через призму КП

Заключение

Итак, следует отметить, что последовательные анализ, выделение главных элементов, а также осуществление возможности планомерного контроля над полнотой связности каждого элемента в контексте проектирования, включая в том числе именование, классификацию и прочие произвольные описания элементов или комментарии, образуют целостную среду моделирования и синтеза структуры алгоритма.

Также следует особо подчеркнуть следующие две кардинальные особенности применения КП:

1. Без КП нет полноты понимания индивидуального качества элементов алгоритма, а также порядка их размещения между смежными элементами проекта.

2. Без КП нет полноты понимания возможностей видоизменения структуры алгоритма, включая масштабирование и агрегацию его частей в структуры иных задач плюс корреляцию функциональности через контроль связей на уровне локального контекста проектирования (ЛКП).

Следует также отметить существующую необходимость в проведении дополнительных исследований возможностей контроля над соответствием описания и содержания элементов КП, а также критериев синтеза структур данных, таких, как способ кодирования и точность представления, которые смогут быть наиболее адекватны для сохранения и визуализации результатов контроля с помощью специальных инструментальных средств.

Литература

1. Miller G.March. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information / G.March // Miller The Psychological Review. – 1956. – № 63(2). – P. 86.
2. Selic B. UML 2: A model-driven development tool / B. Selic // IBM Systems Journal 45. – № 3 (July 1). – P. 607-620.
3. Hungerford Bruce C. The Communication Effectiveness of System Models Using the UML versus Structured Techniques: A Field Experiment. Mid [Электронный ресурс] / Bruce C. Hungerford, Michael A. Eierman // American Journal of Business 20. – 2005. – № 2 (October 1). – P. 35-43. – Режим доступа : http://www.proquest.com/ (accessed September 1, 2007).
4. Пратт Т. Языки программирования: разработка и реализация / Т. Пратт, М. Зелковиц. – СПб., 2002.
5. Аристотель. О памяти и припоминании / Аристотель // Вопросы философии. – 2004. – № 7.
6. Юм Д. Исследование о человеческом разумении / Юм Д. – М. : Прогресс, 1995.
7. Spearman C.E. The abilities of man / Spearman C.E. – N.-Y., 1927.
8. Зинченко П.И. Непроизвольное запоминание / Зинченко П.И. – М., 1961. – 560 с.
9. Смирнов А.А. Проблемы психологии памяти / Смирнов А.А. – М., 1966. – 422 с.
10. Аткинсон Р. Человеческая память и процесс обучения / Аткинсон Р. – М., 1980. – 527 с.
11. Величковский Б.М. Современная когнитивная психология / Величковский Б.М. – М., 1982.
12. Роджерсон Д. Основы СОМ / Роджерсон Д. – М., 1997.

А.І. Розумовський

Реалізація контексту проектування як забезпечення синтезу структури алгоритму

У статті розглядаються ключові проблеми, що виникають при проектуванні складних програмних систем і алгоритмів. Розкрито фактологію стадій проектування, методологічний інструментарій моделювання. Подано практичні рекомендації для побудови локального контексту проектування і створення цілісного середовища моделювання і синтезу структури алгоритму.

Статья поступила в редакцию 09.06.2009.